

**REMARKS**

Claims 1-9 and 11-14 are pending in the application. Claims 1-9 and 11-14 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,338,078 (“Chang”). Applicant requests traversal of these rejections and allowance of the pending claims in view of the following remarks.

Applicants’ invention is directed to a method for controlling the order of datagrams being processed by at least one processing engine. As recited in amended claim 1 for example, the at least one processing engine includes at least one input port and at least one output port, wherein each datagram or each group of datagrams has a ticket associated therewith by a ticket dispenser. Processors in the processing engine, once they become available, interrogate the ticket dispenser to take the next ticket and use it to control the order of the datagram or group of datagrams at the at least one input port of the processing engine and at the at least one output port of the processing engine.

Chang is concerned with preserving packet order in a computer network system. As described (col. 2, lines 41-53), other prior art solutions to queuing inbound packets and distributing them to multiple processors did not take account of scalability nor the preservation of the all-important packet order. The Chang patent makes multiple references to the importance of preserving packet order (see for example, col. 1, lines 52 and 63; col. 2, lines 43-44, 47 and 49; and col. 6, lines 7-12). Chang also indicates (col. 4, lines 56-67 and col. 5, lines 27-47) that single CPU/single thread processor systems suffer in that they are effectively locked out of continuing with their current activity when packets are randomly (i.e. unpredictably) received at

interrupt level. The same is true even of multi-CPU processors operating under single thread operating conditions (col. 5, lines 48-65).

Chang employs multiple CPUs and multiple threads, together with multiple queues, on the basis of one thread and one queue for each CPU, as illustrated in Figure 3. A hashing function (50) is used to allocate threads to queues (62-68) and to CPUs (54-60) on the basis of packets delivered from device drivers (42). Chang indicates that techniques exist for ensuring packets being received from a client are sequenced properly regardless of protocol (col. 6, lines 15+). The method of Chang simply “applies this sequencing in the context of improved parallelization of multiple CPUs in a network system” (col. 6, lines 18-20). Chang does not appear to introduce a new method or technique *per se* for queuing packets to preserve their order but merely applies known techniques to *multithreaded* processor systems, where there is one thread and one queue per processor.

According to one implementation of Chang (col. 6, lines 53 et seq.), the hashing function utilizes MAC addresses applied to packet headers to hash each of the MAC addresses into the plurality of queues so that packets associated with a given device will all be handled in the same particular queue. Chang is said not to be limited to the use of MAC addresses but extends also to other components of the protocol stack (as illustrated in Figure 4 of Chang), none of which appears relevant to Applicant’s invention.

Chang fails to describe how the header address is used in the hashing function to preserve packet order. According to Chang (col. 2, lines 55-67), when a queue is started, a thread is scheduled to process packets on that queue. When all of the packets have been processed, the

thread becomes dormant. The queue waits for the next packet to arrive thus operating in a passive manner.

As described in a previous response, Applicant's invention employs a proactive approach.

In contrast to Chang, Applicant's invention uses a proactive approach. Once a processor becomes available from a previous task, the processor interrogates a ticket allocator (or, dispenser) to take the next ticket associated with a datagram in the incoming queue, thereby to perform an operation of some kind on the datagram to which that ticket relates. The datagram processed by that particular processor may not necessarily be the next datagram in the original numerical sequence since other processors may have taken other tickets in the queue beforehand, depending on the time taken by each processor to perform its operation on the respective datagram. This process is described in Applicant's invention (p. 9, lines 35 - 42). A new processor can join the algorithm operating the present method by simply taking a ticket (p. 8, lines 44 - 47) or a processor can drop out by not interrogating the dispenser.

Applicant's invention increases efficiency and utilization of multiple processors in one or more engines. As opposed to an available processor having to wait to be given a ticket and therefore a data item to process, the method of Applicant's invention actively seeks out a data item. The idle time is negligible in comparison with known deli-counter algorithms. To highlight the distinction, the ticket is sought out by the processor as opposed to the ticket being assigned to the data item. As described by Applicant (pp. 8 and 9, steps 1-10), the first step is where the processor takes a ticket. This is achieved by interrogating the dispenser, once the

processor has become available, to obtain the next available ticket. The processor returns the ticket to the dispenser in next step and reads in a datagram from the source. It is not significant as to which processor operates on which datagram(s).

Chang discloses (in reference to Figure 3 and claims 1, 5 and 8 of Applicant's invention), a processing engine for processing datagrams in a predetermined order, each processing engine having an input port, an output port and a plurality of processing elements (PEs), each PE having an input connected to the input port and an output connected to the output port. The input and output ports are constituted by an input/output port connected to the device drivers in Chang. Chang also discloses arithmetic and logic means.

Chang, however, fails to disclose a ticket dispenser adapted to associate a ticket with each incoming datagram and to be interrogated by the processing elements once they have become available. As recognized in the Office Action, Chang does not disclose a ticket dispenser adapted to associate a ticket with each incoming datagram. However, it is asserted that it would have been obvious to one of ordinary skill in the art to have incorporated the use of a queuing mechanism instead of a ticket dispenser to align packets in certain sequences before being processed by multiple processors in order to make the processing of packets more efficient. There is no suggestion in Chang for such a modification.

Assuming, *strictly arguendo*, that such modification is permissible/envisioned, Chang fails to disclose a ticket dispenser that operates in a proactive manner as described above.

Chang merely provides for packet header addresses to be used for ordering packets in the plurality of queues via the hashing function. As is known, a hashing function merely generates a

(unique) identifier from input data. It has nothing to do with tickets. Applicant's invention, on the other hand, includes at least two further steps. First, the issuing of tickets to processors. As in deli counter operations, the allocated ticket has a bearing on packet order. The disadvantages of the basic deli counter algorithm have been set out in Applicant's specification and are quoted above. Chang sets out the problems of idle time in prior art queuing systems yet his system also results in a dormant queue once processing of the data in a queue has been completed.

Applicant's invention takes a further step by arranging for the ticket dispenser to operate proactively. Rather than a processor (PE) waiting to be presented with the next packet to arrive, Applicant's invention interrogates the ticket dispenser to look for a packet to be processed and to allocate a ticket to it in order to control the order of packets (datagrams) at the input port and at the output port. There is no control of packet order at the input port in Chang since this function is performed by the device drivers.

In Applicant's invention, packets are processed in their turn. A packet that takes longer to process than the average packet may emerge out of order, perhaps even after a later (received) packet. The tickets can be passed to the output side of the processor to preserve order if required. Otherwise, if packet order is not vital, the packets can arrive at the output in the order they emerge from the processors. Chang fails to describe such a feature.

Chang fails to provide, to one of ordinary skill in the art, to consider the first step as an option. Moreover, there is no basis in Chang for the person of ordinary skill in the art to take the further step of devising a ticket dispenser that takes a proactive role. Applicant, therefore, respectfully submits that Chang fails to teach or disclose multiple aspects of Applicant's

invention as claimed. These include, at least: (1) the allocation of tickets; (2) the proactive approach to allocating tickets to processors; and (3) control of packet order at the input port.

Chang leads the person of ordinary skill in the art into accepting that completion of processing of all the packets in a queue automatically leads to that queue becoming dormant. It is not perceived as a problem in Chang, so there would be no reason for the skilled person to look for ways of finding or offering a solution. It would not be within the skills of the ordinary person to invent the proactive step of Applicant's invention. The Office Action has failed to provide support or justification (absent hindsight) for the contention that it would have been obvious to modify Chang in the manner suggested.

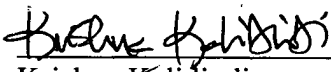
Referring to the assertion in the Office Action that it would have been obvious to one of ordinary skill in the art to have incorporated the use of a queuing mechanism instead of a ticket dispenser to align packets in certain sequences before being processed by multiple processors in order to make the processing of packets more efficient, this assertion does not appear consistent with the claims. The assertion appears to state that the present invention replaces a ticket dispenser with a queuing mechanism. However, Applicant's invention does in fact provide a ticket dispenser. To that extent, the invention does *not* incorporate a queuing mechanism *in place of* a ticket dispenser as stated by the examiner. It, therefore, appears that the Office Action, has misinterpreted the present invention in the opposite sense to the claims.

At least for these reasons, it is believed that claims 1, 5 and 8 are allowable over the teachings of Chang. The remaining claims (i.e. claims 2-4, 6, 7, 9 and 11-14), all of which depend on one of allowable independent claims 1, 5 and 8, are also allowable.

All of the rejections having been overcome, it is believed that this application is in condition for allowance and a notice to that effect is earnestly solicited. Should the Examiner have any questions with respect to expediting the prosecution of this application, she is urged to contact the undersigned at the number listed below.

Respectfully submitted,

Potomac Patent Group, PLLC

By:   
Krishna Kalidindi  
Reg. No. 41,461

Potomac Patent Group PLLC  
P.O. Box 270  
Fredericksburg, VA 22404  
703-893-8500

Date: May 30, 2006